第五章 用户管理功能的实现

本章实验主要完成用户的注册和登录功能的实现。注册功能实现的是将用户的信息放入到数据库的用户表中，如果已经注册过，应该提醒重新选择用户名进行注册；用户登录功能实现的是根据用户登录信息，系统验证用户的合法性，如果验证通过，则将用户的信息显示在页面上，否则就提醒用户，登录的用户名或者密码错误。功能实现使用的 Web 编程语言是 ASP，可以参考文献[4,9,10]进一步了解 ASP 技术和 HTML 网页设计语言。事实上本章涉及到的 ASP 内容很少，主要是 HTML 嵌入少量 VB 代码即可完成，这和 JSP 等 Web 编程语言思路是一样的，只是一些关键词和语法格式有所不同。

第一节 数据库连接的建立

在第三章中确定了二手图书网站的需求分析和总体设计，给出了系统的基本模块功能，其中用户管理模块需要首先实现，该模块包括用户注册和用户登录两部分，而用户登录和用户注册都需要访问数据库，因此可以将数据库连接的功能单独抽取出来，专门负责和数据库建立连接，其它所有访问数据库的模块直接调用这个连接模块就可以了，列表 5.1 中名字为 conn.asp 的网页就是实现数据库连接功能的代码，也可以说是公用模块。其它任何模块如果需要和数据库建立连接，直接使用 include 标签将该 asp 文档插入模块中就可以了，这和一般的函数调用还是不同的，因为这种调用没有参数传递，直接将 conn.asp 中的代码插入调用的文档中，就会节省重复编写代码的时间。另外一个好处就是数据库连接需要修改时，直接修改 conn.asp 中的代码就可以了，否则需要将所有使用数据库连接的 asp 页面都要打开修改，这样容易造成错误，而且增加工作量。这种代码复用的好处应该是显而易见的，没有什么玄奥的机理。

我们实验选择的支撑平台是虚拟主机，对于开发人员而言，仅仅了解列表 5.1 中 datapath 的含义就可以了，也就是需要将用户将放置在虚拟主机逻辑路径下的数据库文件路径写全即可。比如放在虚拟主机中 secondBookStore\这个目录下，数据库文件名为 bookSell.mdb.则此时的数据库文件的路径设置应该为：datapath ="\secondBookStore \bookSell.mdb"。这样 ASP 页面在访问数据库时候，就查找这个路径，找到这个数据库文件，进而实现对数据库中表单的操作。列表 5.1 中 conn.asp 每一行代码几乎都进行了注释，也就是撇号后面的文字说明。这里面还有一条语句需要注意，那就是 Set conn=Server.CreateObject("ADODB.Connection"),这一条语句需要理解的仅仅是 conn 这个返回对象。建立数据库连接的目的就是向数据库系统发送 SQL 语句，因此写好 SQL 字符串，需要调用 conn 这个对象里面的函数将 SQL 语句发送到数据库管理系统并执行，因此使用数据库连接的时候，需要记住这个 conn 对象的名称。当然 conn 这个对象名称是设计人员自己命名的，取任何名称都可以，只要有意义即可，否则用的时候不容易搞清楚是什么东西。如果还想进一步理解其它语句的含义，可能要下点功夫看数据库连接的具体内容和知识了，这在随便一本 ASP，JSP 和其它 web 编程的书籍中都有介绍。如果希望真正理解数据库连接机理，则需要弄清楚其核心思想，这个思想来源于《计算机网络》这门课中的链路建立，也就是建立一个套接字实现服务器监听，客户端根据套接字和服务器建立链璐，进而实现通讯。其实数据库连接就是这个意思，只是将细节封装了，使得开发人员看不到具体实现内容，减轻开发人员的项目开发负担，只要了解如何调用就可以了，因此对一般开发人员而言，也的确没有必要深入了解其细节原理。

列表 5.1：数据库连接模块代码

Conn.asp 文档

```
<%
response.buffer=true '启用缓冲处理。注意 ASP 每一行前面有单引号，表示这行是注释。
dim connstr,datapath,conn,dbpath,sql  'asp 变量可以不声明，声明一下的好处是增加可读性。
datapath="\bookSell.mdb"       '路径开始必须使用符号“\”开始
connstr="Provider=Microsoft.JET.OLEDB.4.0;Data Source=" & Server . mappath ( datapath ) & "; Persist Security Info=False;"    '&的含义，就是将字符串和变量合成一个新字符串
```

```
conn=Server. cCreateObjet("ADODB.Connection")  '创建一个 conn 字符串，set 可以不要
conn.open connstr   '打开数据库链接
' 下面的 if 语句目的是如果链接没有建立起来，则返回数据库建立连接错误。
If Err Then
    err.Clear
    Set Conn = Nothing
   Response.Write "数据库连接出错，请检查数据库连接文件中的数据库参数设置。"
    Response.End
End If
%>
```

第二节 用户注册模块

实现用户的注册功能，显然需要提供用户注册接口的页面，还要有一个负责接收用户的注册数据并将其添加到数据库表里的页面。这样就出现了成对的页面：一个 HTML 负责和用户接口，一个 ASP 负责接收用户发送来的数据，然后和数据库系统配合实现用户注册功能。这就是一个扮演客户端的功能，一个扮演服务器端的功能，其目的就是为了用户远程访问数据库。这和在淘宝等电子商务网站搜索商品一样：先设计制作一个 HTML 页面让用户输入想要搜索的商品名称，然后在服务器端设计编写一个 ASP 页面接收用户 HTML 页面发送过来的这件商品名称，最后到数据库去查找该商品，有的话就将其商品信息返回给用户以供浏览，这也是很自然的过程。现在剩下的问题就是只要理解和掌握 HTML 页面和对应 ASP 页面如何配合工作的就可以了。

现在问题的关键是如何实现两个页面这种对应关系的？也就是一个 HTML 页面怎样才能找到服务器端与之对应的 ASP 页面呢？注意列表 5.1 中的 userRegister.html 以及列表 5.2 中的 userRegister.asp 这两个页面。在 HTML 页面里有<FORM name="form1" method="post" action="**userRegister.asp**">这一行，该行里面 action 后面的这个属性值就是接收本 HTML 页面发来数据的 ASP 页面名称，也就是说接收页面的 ASP 文件名必须作为 HTML 页面中 form 标签 action 的属性值。因为客户端 HTML 页面向服务器端发送数据时，需要 form 这个表单的 action 指定接收的 ASP 页面名称。当然 ASP 页面的文件名称是用户在详细设计中已经取定的，只要保证 action 后面的属性值是接收 ASP 页面的文件名称就可以了。一句话概括说明就是发送数据 HTML 页面和接收数据的 ASP 页面是通过 HTML 页面的 Form 标签关联的。

现在需要解决的第二个问题就是注册页面 userRegister.html 中需要发送的数据可能很多，也就是一次性发送给对应的 ASP 页面 userRegister.asp 可能有很多数据，比如本用户注册 HTML 页面就包含用户名，用户密码和用户手机号，如图 5.1 所示。这些数据是怎样传送给 userRegister.asp 中对应的变量，而且不至于混乱的呢？观察列表 5.1 和 5.2 黑斜体对应的部分。很容易发现列表 5.2 中的 userRegister.asp 代码有 request.form()语句。这个 request 可以在任何 asp 页面中直接使用，它就是专门用来接收客户端 HTML 页面发送来数据的。现在问题是它是如何知道客户端发送来的数据中，哪个数据是用户姓名，哪个是用户密码，哪个是用户手机号码的呢？观察黑斜体很容易发现，这就是依靠语句 request.form("userName")引号中的 useName，它对应 userRegister.html 页面中的 input 语句，也就是标签<input name="***userName***" type="text"> 中的“userName”。比如在 HTML 页面用户名 input 标签输入“张三”这个用户名，那么用户提交注册按钮后，网络就会将“userName：张三”组成一对发送给 useRegister.asp 页面，于是这个 ASP 页面就可以使用语句 **uName** = request.form("***userName***")，将 HTML 页面发来的 userName 对应的“张三”存放到 uName 这个变量里面了，也就是 uName 变量存放一个字符串“张三”。需要说明的是：uName 也可以取名为 userName,但是它与 HTML 中 input 标签里面的“userName”本质完全不一样， uName(当然也可以命名为 userName)是 ASP 页面里面的 VB 语言变量，而"userName"是字符串常量，其目的是为了保持 html 中提交的用户名和 ASP 页面中 request.form("userName")一一对应起来，以便于 ASP 页面知道 request.form("userName")返回的是什么，取出这个返回值然后赋给变量 uName。既然 uName 是 ASP 页面中的 VB 变量，它的名称可以随便取，取名 userName 当然也可以，显然 userName 和"userName"不是一回事，没有任何关系，一个是变量名，一个是字符串常量。这样 ASP 通过三条 request.form()语句，使用三个字符串，就将客户端 HTML

发来的用户名，用户密码和用户手机号逐一取出来，放入 ASP 的对应的 VB 变量中，以备后面代码的调用。

现在第三个问题来了。就是用户注册功能。在网站数据库设计实验中规定需要将数据放到数据库的用户注册表里面的，那么现在怎样将拿到的用户名，用户密码和用户手机号码放到这个表中去呢？显然需要在 userRegister.asp 页面中实现。观察该页面中的语句：

```
mySQL="insert into userInfo(userName,userPassword,userPhone)values('"&uName&"','"&uPassword&"','"&uPhone&"')"
```

理解这条语句，就能基本解决了 web 访问数据库的编程问题了。这是一个字符串，就是一个 SQL 插入语句字符串，实现将用户名，用户密码和用户手机号码插入到数据库的表 userInfo 中。学过数据库管理系统的同学都应该熟悉这条语句。现在 ASP 必须生成这条语句，以字符串的形式传输给数据库管理系统，然后由数据库管理系统来具体执行。既然是 SQL 语句，当然需要遵守 SQL 语法结构，那就是 insert into 是执行插入命令的关键词，userInfo 是具体的存放用户数据的表，(userName, userPassword, userPhone)是数据库 userInfo 这个表中的字段变量，这都是在第四章的数据库设计实验完成的，也就是数据库的 userInfo 表已经创建好了，现在是具体使用了。该 SQL 语句中的 values()具体包含有用户名、用户密码和用户手机号，它们是变量，因此不能直接写在字符串里面。例如这三个变量具体存放的分别是用户从 HTML 发来的数据：“张三”，“888888”，“13776785423”。现在要生成 SQL 语句，只能使用 VB 语言的&运算符号将变量值和固定的 SQL 语句其它部分串联起来，形成一个完整的 SQL 语句。这就很容易明白('"&uName &"','"&uPassword&"','"&uPhone&"')的含义了。这里面有单引号和双引号，还有逗号，具体原因是 SQL 语句的 insert 插入语句应该是

```
insert into userInfo(userName,userPassword,userPhone) values ('张三','888888','13776785423')
```

如果要生成这种长字符串，也就是 values 后面的每一个具体变量值需要单引号引起来，而且还需要使用西文逗号将它们分开，也只能写成('"&uName &"','"&uPassword&"','"&uPhone&"')，因为计算机执行的时候就会将 uName,uPassword,uPhone 的值代替这三个变量名称进而生成字符串。另外需要注意 userInfo(userName,userPassword,userPhone)中，userInfo 是表的名称，userName, userPassword, userPhone 是表的字段变量名称，它们都是作为字符串的一部分，和 HTML 页面以及 ASP 中其它 VB 变量没有任何关系。当然数据库系统里面的字段变量 userName 也可以命名为 uName，但是和 ASP 页面里面的 uName 没有任何关系，分属于不同的体系。比如现在某个班级里面有个同学的名字叫周杰伦，他和歌星周杰伦没有任何关系，何况字段变量在 ASP 页面里是以字符串形式出现的，目的仅仅是为了生成 SQL 语句。最后强调一下，字段变量名称不能乱取，不能和数据库系统中的关键词重名，比如 userInfo 表有用户密码字段变量，不能取名为 password，因为数据库系统中有保留字，就是 password，也不能使用 user 作为字段变量的名称或者表的名称。

现在需要说明一下：在编写注册和登录程序之前，需要将数据库设计和创建好，这是实验四需要完成的，有的同学并没有完成，因此无法访问数据库了。另外就是列表 5.1 和列表 5.2 代码仅仅是示例。在实验四中设计的注册用户信息可能很多，需要依葫芦画瓢，改造图 5.1 所示的 HTML 页面，增加用户的属性信息 input 标签，当然也要相应地改造列表 5.2 所示的 ASP 页面，增加 request.form()以接收增加的注册数据，然后再改造 SQL 语句，最后实现将所有数据插入到表中。

如果认真学习和实验过 ASP 或者 JSP 等 web 编程语言，则不需要上面那么多的详细说明，只要上机实验编写一个使用 java 或者 c 语言的小系统，比如完成一个简单的聊天软件，或者编写一个类似计算器的小程序都可以迅速掌握上面知识的。

用户的注册和登录功能已经基本讲解完了，剩下的还要理解列表 5.2 中的 conn.Execute mySQL 语句，这条语句功能的就是调用<!-- #include file="conn.asp" -->中的 conn.asp 创建的 conn 这个对象，然后执行这个对象的函数 Execute()，实现将生成的 mySQL 字符串发送到 access 数据库管理系统，执行这条 insert 语句。具体实现细节，ASP 服务器都做了很好的封装，用户只需要调用即可。至于后面的 conn.close 等语句，学习过 Java 的同学都了解，就是删除建立的对象，释放资源。

userRegister.html 页面最后有<a href=UserInfoList.asp>这个标签语句，我们的目的是检验注册的数据是否添加到数据库的 userInfo 表中。商用的系统是不可能加上这条语句的，否则就麻烦了，因为该 ASP 页面的执行泄露了所有注册用户的信息。

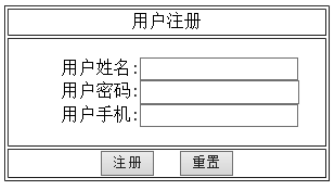

图 5.1 用户注册页面

列表 5.1：客户端用户注册页面(userRegister.html)

```
<html>
<body>
<DIV align="center">
  <TABLE width="300" border="1" align="center" bordercolor="#000000">
    <FORM name="form1" method="post" action="userRegister.asp">
      <TR>  <TD height="25" colspan="6" align="center" class="tr">用户注册</TD>   </TR>
      <TR>  <TD height="100" colspan="6" align="center" >
        <TABLE width="96%" border="0" align="center" cellpadding="0" cellspacing="0">
          <TR> <TD width="41%" align="right">用户姓名:</TD> <TD width="59%"><input name="userName" type="text"></TD>
          </TR> <TD align="right">用户密码:</TD>  <TD><input name="userPassword" type="password"></TD> </TR>
        <TR><TD width="41%" align="right">用户手机:</TD><TD width="59%"><input name="userPhone" type="text"></TD> </TR>
        </TABLE>
        </TD> </TR>
      <TR> <TD height="25" colspan="6" align="center" class="tr">
           <INPUT type="submit" name="Submit1" value="注册">  <INPUT type="reset" name="Submit2" value="重置"></TD>
      </TR>
    </FORM>
  </TABLE>
</DIV>
</body>
</html>
```

列表 5.2：服务器用户注册页面(userRegister.asp)

```
<!-- #include file="conn.asp" -->  'include 就是将 conn.asp 内容插入进来，实现数据库连接
<html>
<head>
    <title>实现数据库的数据插入</title>
</head>
<body>
<%
     uName   =  request.form("userName")
     uPassword  =  request.form("userPassword")
     uPhone  =  request.form("userPhone")
     mySQL="insert into userInfo(userName,userPassword,userPhone)values('"&uName&"','"&uPassword&"','"&uPhone&"')"
    conn.Execute mySQL
    conn.close
    set conn=nothing
%>
   祝贺你，注册成功
   <a href=UserInfoList.asp>查看当前注册学生信息</a>"
</body>
</html>
```

第三节 用户登录模块

本节完成实验三中用户的登录功能的页面开发。实验三中设计了用户登录的流程，基本步骤就是用户提交用户名和密码,系统接收这两个数据,然后到数据库的用户信息表中查询是否有这个用户的信息，如果提交的用户名和密码在数据库中查询到，那么登录就成功了，接着就让系统记住该登录用户名，也就是使用内置对象 session 保留登录的用户名，这样其它页面都可以共享使用这个用户名了。比如如果这个用户是卖家，那么在发布图书的时候，系统会自动从 session 里面取出这个用户名，也就是卖家名

称，然后将卖家名称和图书的名称，价格等等数据一起插入到数据库的图书表。显然我们知道用户注册的用户名也就是用户登录的登录名实质是一致的，因此用户信息表中应该是关键字。下面分析具体实现用户登录的代码。

列表 5.3 是用户登录页面 Login.html 的代码，图 5.2 是代码执行后的用户登录 HTML 页面。这个登录页面有用户名和密码两个文本框以及提交按钮。对应的服务器端 ASP 页面即 login.asp 页面，它们之间的数据传递关系，在上一节用户注册功能的实现中讲述得十分清楚了。也就是在 login.asp 中将接收用户名和密码的 VB 变量命名为 userN 和 psw，使用 request.form()获取客户端 login.html 发来的用户登录数据。和用户注册不同的是接收到这两个数据，比如用户名为张三，密码为 8888,下一步不是生成 insert 插入 SQL 语句，而是需要使用 select 语句，到数据库 userInfo 表中查询有没有姓名为张三，密码为 8888 的记录，有的话就返回登录成功，没有的话就登录失败。这种判断有无的依据就是测试返回的记录集合是否为空，如果为空就说明没有满足条件的记录，那么结果就是登录失败，此时用户输入的密码或者用户名要么输入错误，要么这个用户还没有注册。显然实现这种检索记录的 SQL 语句应该是

```
Select * from userInfo where userName = '张三' and  userPassword = '888888'
```

这就意味着我们需要使用 VB 语言生成上面的字符串，同时注意具体生成字符串的时候，张三和 888888 要使用变量 userN 和 psw 代替，同时使用 VB 的字符串连接符号，这和用户注册生成 insert 语句完全一样，如列表 5.3 所示。该列表中最重要的语句就是生成 select 语句的字符串

```
commandStr = "select * from userInfo where userName='" & userN &"' and "&"psw='" &userPassword&"'"
```

用户注册就是将用户注册数据插入数据库的 userInfo 表就结束了，但是用户登录在检索完以后，需要判断返回结果，也就是记录集合是否为空。具体代码语句 set rs = conn.Execute(commandStr)，返回一个对象是 rs，当然返回的这个 rs 的变量名是我们自己取的，系统会将返回的对象集合自动赋给 rs 这个变量，现在剩下的就是判断这个对象集合是否有记录，具体判断就是 rs.EOF，它要么返回 TRUE，要么返回 FALSE。EOF 是 end of file 的首字母缩写，它的意思就是 rs 这个集合的第一行是不是结尾标识，第一行就是结尾标识，就意味着没有数据。如果有一条记录，那么结尾标识就会在第二行，因此如果 rs.EOF 返回的是 TRUE，就表明是第一行就是结尾标识符，因此 if not rs.EOF，这个判断语句如果成立，就表明 rs 这个集合有数据，因此登录成功，否则就是登录失败，具体实现代码见列表 5.3。既然登录成功，我们将在 if 语句后面生成欢迎登录提示语，然后调用 session 内置对象保留登录用户名，也就是 session("username") = userN。注意 session 里面的字符串是开发者自己命名的，右边的 userN 是变量，目前存放的是登录用户名“张三”。其它页面要共享这个用户名的时候，就可以直接将 session("username")赋给一个变量即可。

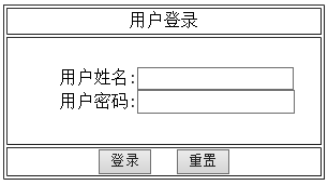

图 5.2 用户登录页面

列表 5.3：客户端用户登录页面(userLogin.html)

```
<html>
<body>
<DIV align="center">
  <TABLE width="300" border="1" align="center" bordercolor="#000000">
    <FORM name="form1" method="post" action="userLogin.asp">
      <TR> <TD height="25" colspan="6" align="center" >用户登录</TD> </TR>
      <TR> <TD height="100" colspan="6" align="center" >
           <TABLE width="96%" border="0" align="center" cellpadding="0" cellspacing="0">
             <TR> <TD width="41%" align="right">用户名:</TD>
                  <TD width="59%"><input name="userName" type="text"></TD>
             </TR>
             <TR> <TD align="right">密码:</TD>
                  <TD><input name="passWord" type="password"></TD>
             </TR>
           </TABLE>
```

```
        </TD>
      </TR>
      <TR>
        <TD height="25" colspan="6" align="center" class="tr">
              <INPUT type="submit" name="Submit1" value="登录">  
              <INPUT type="reset" name="Submit2" value="重置">
        </TD>
      </TR>
    </FORM>
  </TABLE>
</DIV>
</body>
</html>
```

列表 5.4：服务器用户登录页面(login.asp)

```
<!-- #include file="conn.asp" -->
<%
    userN = request.form("username")
    psw   = request.form("password")
    commandStr = "select * from userInfo where userName='" & userN &"'and "&"userPassword='" &psw&"'"
    rs = conn.Execute(commandStr)
    'Response.write(commandstr)
    'rs 返回满足条件的记录，如果有记录说明用户登录成功，没有则说明用户没有注册或者用户名或者密码输入错误
    if not rs.EOF then '如果记录不为空，rs.EOF 返回的是 false，所以 not rs.EOF 就是 true.
        Response.Write "欢迎"&userN&"登录"
        session("username") = userN  '保存登陆的用户名供其它页面共享使用
    else
        Response.write("用户名或者密码错误，请重新登录")
     end if
    conn.close
    Set conn = nothing
%>
```

第四节 一些常见的错误以及解决办法

1.页面汉字是乱码

解决办法：将下面的语句拷贝到页面的第一行，哪个解决问题拷贝哪个，注意拷贝以后，尤其是下面第二个，可能运行以后还是乱码，这时候打开页面，可能里面的汉字已经是乱码了，修改过来即可，有时候要等一会，等待系统刷新以后修改的才能真正起作用。

```
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
```

2. Microsoft JET Database Engine 错误 '80004005' 找不到文件 'd:\host\.....

解决方法：有下面两种可能：(1)没有数据库文件或者创建的数据库文件损坏了，需要创建数据库文件，并将它拷贝到虚拟主机和访问数据库的那个页面相同的目录下面；如果已经创建好了直接拷贝就行了;(2)在 conn.asp 中，datapath="数据库名称.mdb"，这个“数据库名称”和创建的数据库名称不一致，需要修改为创建的数据库名称或者设置正确该数据库文件所在路径。

3. 问题：Microsoft JET Database Engine 错误 '80040e37'。Microsoft Jet 数据库引擎找不到需要插入或查询的表。确定 SQL 语句中的表是否存在，以及它的名称的拼写是否正确。

解决方法：asp 页面中 select * from <表名>，这个表的名称在 select 语句中和数据库表的名称不一致，修改一致即可。

4. 问题：Microsoft JET Database Engine 错误 '80040e10' 至少一个参数没有被指定值

解决方法：asp 接收 html 发送来的 request.form 语句，这个语句中的变量名也就是下面这条语句中的 name=request.form(“username”),name 和生成 SQL 语句中的变量中的 name 一致,也就是所讲的数据发送和接收对应的名称错了。

5. 禁止数据库系统中的数据库名称，字段变量名称使用英文单词，比如使用 user，password 等作为表的名称或者字段变量名称,可以在单词前面加上一两个字符,因为它们是数据库管理系统的保留字。这种错误常常导致很难确定是什么原因。

6. 建议不要使用 accdb 版本的数据库文件，使用 access 转换为 2003 版本的 mdb 数据库文件

7. <!-- #include file="conn.asp" --> 中的!—这个注释必不可少,否则会在页面显示出来

8. vb 语句中的字符串必须使用西文的双引号和单引号，比如字符串使用中文的字符串，就会保错误信息。

9. 设计的数据库的用户表，如果设置了主键，比如用户名设置为主键，那么在注册页面中，如果用户名对应的文本框为空，或者和已经有注册的用户名重名，这时候系统会报错。为了实验方便可以在数据库表设计的时候不设置主键。

第五节 综合实验五

一、实验目的和要求

1 掌握用户注册和登录的实现。

2 掌握查询语句和插入语句的格式和用途。

二、实验环境与工具软件

1.互联网络环境

2.虚拟主机服务器

3.FTP 工具软件

三、实验内容和步骤

1.用户注册

(1)设计注册 html 页面

(2)设计接收 html 发送来用户信息的 ASP 页面

(3)在该 ASP 页面中接收用户信息，并检测注册用户名是否使用过

(4)将用户信息插入到数据库中。

(5)成功插入用户数据后给出提示信息

2.用户登录

(1)设计登录 html 页面

(2)设计接收 html 发送用户登录信息的 ASP 页面

(3)在该 ASP 页面中接收用户信息，并验证是否是合法用户

(4)如果是新用户要求重新注册。

(5)如果是合法用户，保留登录用户信息，并给出欢迎该用户的提示语。

四、实验报告(共 5 分)

1. 写出实现用户管理的代码文件名称以及需要的公共模块代码并说明其功能(1 分)

2. 阅读用户注册代码，测试用户注册页面，并对实现该功能的关键代码给以注释说明 (2 分)

3. 阅读用户登录代码，测试用户登录页面，并对实现该功能的关键代码给以注释说明 (2 分)